1.   A motion buffer, implemented on a machine readable medium, comprising a data structure configured to store on a per pixel basis the local properties of one or more 3-D objects to be composited to a 2-D scene including each 3-D object's color, depth, coverage, transfer mode, and one or more of each 3-D object's rate of change of depth, and surface geometry information.

2.   The motion buffer of claim 1, wherein the surface geometry information for each of the one or more 3-D objects stored in the motion buffer includes per pixel information about the local orientation of the surface of each of the one or more 3-D objects

3.   The motion buffer of claim 1, wherein the data structure is further configured to store the local properties of the one or more 3-D objects in depth sorted order.

4.   The motion buffer of claim 1, wherein the data structure is configured to store the local properties of the one or more 3-D objects as a plurality of linked lists, wherein each linked list corresponds to a pixel in the 2-D scene, and each link in a linked list stores the local properties of one of the one or more 3-D objects.

5.   The motion buffer of claim 4, wherein each link in a linked list comprises a pixel fragment configured to store the local color, depth, coverage and transfer mode of one of the one or more 3-D objects, and one or more of that 3-D object's rate of change of depth, and surface geometry information.

6.   A method for creating a motion buffer to store the local properties of one or more 3-D objects, comprising:

        receiving one or more 3-D objects, wherein each 3-D object comprises one or more object primitives;

        scan-converting each 3-D object's one or more object primitives into a plurality of pixel fragments corresponding to a plurality of pixels in a 2-D scene,

wherein each pixel fragment is configured to store the local properties of a scan-converted object primitive including the object primitive's local color, depth, coverage, and transfer mode, and one or more of the object primitive's local rate of change of depth, and surface geometry information; and

inserting each of the pixel fragments into the motion buffer.

7. The method of claim 6, further comprising inserting each of the pixel fragments into the motion buffer in depth sorted order.

8. The method of claim 6, further comprising storing the motion buffer as a plurality of linked lists corresponding to a plurality of pixels in the 2-D scene, wherein each link in a linked list comprises a pixel fragment having a pointer to the next pixel fragment, if any, in the linked list.

9. A method for compositing one or more 3-D objects to a 2-D scene, comprising:

receiving a motion buffer, the motion buffer containing the rendered local properties of the one or more 3-D objects including each 3-D object's color depth, coverage, transfer mode, and one or more of each 3-D object's rate of change of depth and surface geometry information; and

resolving the motion buffer to composite the one or more 3-D objects to the 2-D scene.

10. The method of claim 9, wherein the step of resolving the motion buffer to composite the one or more 3-D objects to the 2-D scene further comprises blending, on a per pixel basis and in depth sorted order, the color of each of the one or more 3-D objects to the color in the 2-D scene using the transfer mode of each of the one or more 3-D objects.

18

11.   The method of claim 9, wherein the motion buffer contains surface geometry information for each of the one or more 3-D objects and the step of resolving the motion buffer further comprises using the surface geometry information to simultaneously anti-alias the one or more 3-D objects while compositing the one or more 3-D objects to the 2-D scene.

12.   The method of claim 11, wherein two or more of the 3-D objects intersect over an output buffer pixel in the 2-D scene, further comprising:

determining the number of regions in the output buffer pixel in which the one or more intersecting 3-D objects are uniquely layered, and the relative coverage of each uniquely layered region;

determining a blended color for each uniquely layered region by blending in depth sorted order the color of each of the one or more 3-D objects with the color of the output buffer pixel according to each 3-D object's transfer mode; and

painting the output buffer pixel with a weighted average of the blended colors determined for each uniquely layered region, wherein the weight assigned to the blended color of a uniquely layered region is determined by the relative coverage of that region.

13.   The method of claim 9, wherein the motion buffer contains surface geometry information for the one or more 3-D objects and the step of resolving the motion buffer further comprises simultaneously depth-of-field blurring the one or more 3-D objects while compositing the one or more 3-D objects to the 2-D scene.

14.   The method of claim 13, wherein the step of simultaneously depth-of-field blurring the one or more 3-D objects while compositing the one or more 3-D objects to the 2-D scene further comprises:

using the depth and surface geometry information for the one or more 3-D objects to extend, on an output buffer pixel basis, the surfaces of the one or more 3-D objects into an extended output buffer pixel;

19

determining whether the extended surfaces of two or more of the 3-D objects intersect over the extended output buffer pixel; and

blending the colors of the one or more 3-D objects with the color of the output buffer pixel as if two or more of the 3-D objects intersected over the output buffer pixel whenever the extended surfaces of two or more of the 3-D objects intersect over the extended output buffer pixel.

15.   The method of claim 9, wherein the motion buffer contains the surface geometry information for the one or more 3-D objects and the step of resolving the motion buffer further comprises using the the surface geometry information to simultaneously anti-alias and depth-of-field blur the one or more 3-D objects while compositing the one or more 3-D objects to the 2-D scene.

16.   The method of claim 9, wherein the motion buffer contains the rate of change of depth for each of the one or more 3-D objects, and the step of resolving the motion buffer further comprises using the rate of change of depth for each of the one or more 3-D objects to simultaneously motion-blur the one or more 3-D objects while compositing the one or more 3-D objects to the 2-D scene.

17.   The method of claim 16, wherein the surfaces of two or more of the 3-D objects pass through each other over an output buffer pixel in the 2-D scene during a shutter interval, further comprising:

determining the number of time periods during the shutter interval in which the one or more 3-D objects are uniquely layered, and the duration of each uniquely layered time period;

determining a blended color for each uniquely layered time period by blending in depth sorted order the color of each of the one or more 3-D objects with the color of the output buffer pixel according to each of the one or more 3-D objects' transfer modes; and

painting the output buffer pixel with a weighted average of the blended colors for each uniquely layered time period, wherein the weight assigned to the blended color of a uniquely layered time period is determined by the duration of that time period.

18.   The method of claim 9, wherein the motion buffer contains the rate of change of depth and surface geometry information for the one or more 3-D objects, and the step of resolving the motion buffer further comprises using the rate of change of depth and surface geometry information for the one or more 3-D objects to simultaneously anti-alias and motion-blur the one or more 3-D objects while compositing the one or more 3-D objects to the 2-D scene.

19.   The method of claim 18, wherein the surfaces of two or more of the 3-D objects intersect and pass through each other over an output buffer pixel in the 2-D scene during a shutter interval, further comprising:

        dividing the area of the output buffer pixel and the shutter interval into a number of uniquely layered space-time regions, wherein for each uniquely layered space-time region the surfaces of the one or more 3-D objects are uniquely layered over a portion of the output buffer pixel for a portion of the shutter interval;

        determining the number and volume of each uniquely layered space-time region, wherein the volume of a uniquely layered space-time region is calculated from the portion of the output buffer pixel and the portion of the shutter interval occupied by the space-time region;

        determining a blended color for each uniquely layered space-time region by blending in depth sorted order the color of each of the one or more 3-D objects stored in the motion buffer with the color of the output buffer pixel according to each object's transfer mode; and

        painting the output buffer pixel with a weighted average of the blended colors for each uniquely layered space-time region, wherein the weight assigned to the

blended color of a uniquely layered space-time region is determined by the volume of that uniquely layered space-time region.

20.   The method of claim 9, wherein the motion buffer contains the rate of change of depth and surface geometry information for the one or more 3-D objects, and the step of resolving the motion buffer further comprises using the rate of change of depth and surface geometry information for the one or more 3-D objects to simultaneously motion-blur and depth-of-field blur the one or more 3-D objects while compositing the one or more 3-D objects to the 2-D scene.

21.   The method of claim 9, wherein the motion buffer contains the rate of change of depth and surface geometry information for the one or more 3-D objects, and the step of resolving the motion buffer further comprises using the rate of change of depth and surface geometry information for the one or more 3-D objects to simultaneously anti-alias, motion-blur and depth-of-field blur the one or more 3-D objects while compositing the one or more 3-D objects to the 2-D scene.

22.   A method for rendering a plurality of 3-D objects to a 2-D scene, comprising:

    splitting the plurality of 3-D objects into one or more non-interacting object clusters;

    rendering all non-simple and non-interacting object clusters to a motion buffer; and

    resolving the motion buffer to composite the non-simple and non-interacting object clusters to the 2-D scene.

23.   The method of claim 22, further comprising rendering all simple and non-interacting object clusters directly to the 2-D scene.

22

24.   The method of claim 22, further comprising rendering all simple and non-interacting object clusters to the motion buffer; and resolving the motion buffer to composite both the simple and non-interacting object clusters and the non-simple and non-interacting object clusters to the 2-D scene.

25.   The method of claim 22, further comprising creating a merged motion buffer by adding to the contents of the motion buffer the contents of a second motion buffer containing one or more separately rendered 3-D objects; and compositing the contents of the merged motion buffer to the 2-D scene.

26.   A computer program product, implemented on a machine readable medium, for creating a motion buffer to store the local properties of one or more 3-D objects, the computer program product comprising instructions operable to cause a programmable processor to:

        receive one or more 3-D objects, wherein each 3-D object comprises one or more object primitives;

        scan-convert each 3-D object's one or more object primitives into a plurality of pixel fragments corresponding to a plurality of pixels in a 2-D scene, wherein each pixel fragment is configured to store the local properties of a scan-converted object primitive including the object primitive's local color, depth, coverage, and transfer mode, and one or more of the object primitive's local rate of change of depth, and surface geometry information; and

        insert each of the pixel fragments into the motion buffer.

27.   The computer program product of claim 26, further comprising instructions operable to cause a programmable processor to insert each of the pixel fragments into the motion buffer in depth sorted order.

28.   The computer program product of claim 26, further comprising instructions operable to cause a programmable processor to store the motion buffer as a plurality

23

of linked lists corresponding to a plurality of pixels in the 2-D scene, wherein each link in a linked list comprises a pixel fragment having a pointer to the next pixel fragment, if any, in the linked list.

29.   A computer program product, implemented on a machine readable medium, for compositing one or more 3-D objects to a 2-D scene, the computer program product comprising instructions operable to cause a programmable processor to:

receive a motion buffer, the motion buffer containing the rendered local properties of the one or more 3-D objects including each 3-D object's color depth, coverage, transfer mode, and one or more of each 3-D object's rate of change of depth and surface geometry information; and

resolve the motion buffer to composite the one or more 3-D objects to the 2-D scene.

30.   The computer program product of claim 29, wherein the instructions to resolve the motion buffer to composite the one or more 3-D objects to the 2-D scene further comprises instructions to blend, on a per pixel basis and in depth sorted order, the color of each of the one or more 3-D objects to the color in the 2-D scene using the transfer mode of each of the one or more 3-D objects.

31.   The computer program product of claim 29, wherein the motion buffer contains surface geometry information for each of the one or more 3-D objects and the instructions to resolve the motion buffer further comprises instructions to use the surface geometry information to simultaneously anti-alias the one or more 3-D objects while compositing the one or more 3-D objects to the 2-D scene.

32.   The computer program product of claim 31, wherein two or more of the 3-D objects intersect over an output buffer pixel in the 2-D scene, further comprising instructions operable to cause a programmable processor to:

24

determine the number of regions in the output buffer pixel in which the one or more intersecting 3-D objects are uniquely layered, and the relative coverage of each uniquely layered region;

determine a blended color for each uniquely layered region by blending in depth sorted order the color of each of the one or more 3-D objects with the color of the output buffer pixel according to each 3-D object's transfer mode; and

paint the output buffer pixel with a weighted average of the blended colors determined for each uniquely layered region, wherein the weight assigned to the blended color of a uniquely layered region is determined by the relative coverage of that region.

33.   The computer program product of claim 29, wherein the motion buffer contains surface geometry information for the one or more 3-D objects and the instructions to resolve the motion buffer further comprises instructions to simultaneously depth-of-field blur the one or more 3-D objects while compositing the one or more 3-D objects to the 2-D scene.

34.   The computer program product of claim 33, wherein the instructions to simultaneously depth-of-field blurr the one or more 3-D objects while compositing the one or more 3-D objects to the 2-D scene further comprises instructions to:

use the depth and surface geometry information for the one or more 3-D objects to extend, on an output buffer pixel basis, the surfaces of the one or more 3-D objects into an extended output buffer pixel;

determine whether the extended surfaces of two or more of the 3-D objects intersect over the extended output buffer pixel; and

blend the colors of the one or more 3-D objects with the color of the output buffer pixel as if two or more of the 3-D objects intersected over the output buffer pixel whenever the extended surfaces of two or more of the 3-D objects intersect over the extended output buffer pixel.

25

35.   The computer program product of claim 29, wherein the motion buffer contains the surface geometry information for the one or more 3-D objects and the instructions to resolve the motion buffer further comprises instructions to use the the surface geometry information to simultaneously anti-alias and depth-of-field blur the one or more 3-D objects while compositing the one or more 3-D objects to the 2-D scene.

36.   The computer program product of claim 29, wherein the motion buffer contains the rate of change of depth for each of the one or more 3-D objects, and the instructions to resolve the motion buffer further comprises instructions to use the rate of change of depth for each of the one or more 3-D objects to simultaneously motion-blur the one or more 3-D objects while compositing the one or more 3-D objects to the 2-D scene.

37.   The computer program product of claim 36, wherein the surfaces of two or more of the 3-D objects pass through each other over an output buffer pixel in the 2-D scene during a shutter interval, further comprising instructions operable to cause the programmable processor to:

determine the number of time periods during the shutter interval in which the one or more 3-D objects are uniquely layered, and the duration of each uniquely layered time period;

determine a blended color for each uniquely layered time period by blending in depth sorted order the color of each of the one or more 3-D objects with the color of the output buffer pixel according to each of the one or more 3-D objects' transfer modes; and

paint the output buffer pixel with a weighted average of the blended colors for each uniquely layered time period, wherein the weight assigned to the blended color of a uniquely layered time period is determined by the duration of that time period.

38.   The computer program product of claim 29, wherein the motion buffer contains the rate of change of depth and surface geometry information for the one or more 3-D objects, and the instructions to resolve the motion buffer further comprises instructions to use the rate of change of depth and surface geometry information for the one or more 3-D objects to simultaneously anti-alias and motion-blur the one or more 3-D objects while compositing the one or more 3-D objects to the 2-D scene.

39.   The computer program product of claim 38, wherein the surfaces of two or more of the 3-D objects intersect and pass through each other over an output buffer pixel in the 2-D scene during a shutter interval, further comprising instructions operable to cause the programmable processor to:

divide the area of the output buffer pixel and the shutter interval into a number of uniquely layered space-time regions, wherein for each uniquely layered space-time region the surfaces of the one or more 3-D objects are uniquely layered over a portion of the output buffer pixel for a portion of the shutter interval;

determine the number and volume of each uniquely layered space-time region, wherein the volume of a uniquely layered space-time region is calculated from the portion of the output buffer pixel and the portion of the shutter interval occupied by the space-time region;

determine a blended color for each uniquely layered space-time region by blending in depth sorted order the color of each of the one or more 3-D objects stored in the motion buffer with the color of the output buffer pixel according to each object's transfer mode; and

paint the output buffer pixel with a weighted average of the blended colors for each uniquely layered space-time region, wherein the weight assigned to the blended color of a uniquely layered space-time region is determined by the volume of that uniquely layered space-time region.

40.   The computer program product of claim 29, wherein the motion buffer contains the rate of change of depth and surface geometry information for the one or

more 3-D objects, and the instructions to resolve the motion buffer further comprises instructions to use the rate of change of depth and surface geometry information for the one or more 3-D objects to simultaneously motion-blur and depth-of-field blur the one or more 3-D objects while compositing the one or more 3-D objects to the 2-D scene.

41.   The computer program product of claim 29, wherein the motion buffer contains the rate of change of depth and surface geometry information for the one or more 3-D objects, and the instructions to resolve the motion buffer further comprises instructions to use the rate of change of depth and surface geometry information for the one or more 3-D objects to simultaneously anti-alias, motion-blur and depth-of-field blur the one or more 3-D objects while compositing the one or more 3-D objects to the 2-D scene.

42.   A computer program product, implemented on a machine readable medium, for rendering a plurality of 3-D objects to a 2-D scene, the computer program product comprising instructions operable to cause a programmable processor to:

        split the plurality of 3-D objects into one or more non-interacting object clusters;

        render all non-simple and non-interacting object clusters to a motion buffer; and

        resolve the motion buffer to composite the non-simple and non-interacting object clusters to the 2-D scene.

43.   The computer program product of claim 42, further comprising instructions operable to cause the programmable processor to render all simple and non-interacting object clusters directly to the 2-D scene.

44.   The computer program product of claim 42, further comprising instructions operable to cause the programmable processor to render all simple and non-

interacting object clusters to the motion buffer; and to resolve the motion buffer to composite both the simple and non-interacting object clusters and the non-simple and non-interacting object clusters to the 2-D scene.

45.   The computer program product of claim 42, further comprising instructions operable to cause the programmable processor to create a merged motion buffer by adding to the contents of the motion buffer the contents of a second motion buffer containing one or more separately rendered 3-D objects; and compositing the contents of the merged motion buffer to the 2-D scene.